

Online Voter Control in Sequential Elections*

Edith Hemaspaandra[†]
 Department of Computer Science
 Rochester Institute of Technology
 Rochester, NY 14623, USA

Lane A. Hemaspaandra[‡]
 Department of Computer Science
 University of Rochester
 Rochester, NY 14627, USA

Jörg Rothe[§]
 Institut für Informatik
 Heinrich-Heine-Universität Düsseldorf
 40225 Düsseldorf, Germany

March 2, 2012

Abstract

Previous work on voter control, which refers to situations where a chair seeks to change the outcome of an election by deleting, adding, or partitioning voters, takes for granted that the chair knows all the voters' preferences and that all votes are cast simultaneously. However, elections are often held sequentially and the chair thus knows only the previously cast votes and not the future ones, yet needs to decide instantaneously which control action to take. We introduce a framework that models *online voter control in sequential elections*. We show that the related problems can be much harder than in the standard (non-online) case: For certain election systems, even with efficient winner problems, online control by deleting, adding, or partitioning voters is PSPACE-complete, even if there are only two candidates. In addition, we obtain completeness for coNP in the deleting/adding cases with a bounded deletion/addition limit, and for NP in the partition cases with only one candidate. Finally, we show that for plurality, online control by deleting or adding voters is in P, and for partitioning voters is coNP-hard.

1 Introduction

The study of the computational properties of voting systems has been an exceedingly active area within computational social choice. In particular, various types of manipulation, electoral control, and bribery

*Also appears as URCS-TR-2012-976.

[†]Supported in part by grant NSF-CCF-1101452 and a Friedrich Wilhelm Bessel Research Award. Work done in part while visiting Heinrich-Heine-Universität Düsseldorf.

[‡]Supported in part by grants NSF-CCF-{0426761,0915792,1101479} and arc-dp110101792, and a Friedrich Wilhelm Bessel Research Award. Work done in part while visiting Heinrich-Heine-Universität Düsseldorf.

[§]Supported in part by DFG grant RO-1202/15-1, SFF grant "Cooperative Normsetting" of HHU Düsseldorf, and a DAAD grant for a PPP project in the PROCOPE program.

in voting have been classified in terms of their computational complexity (see [FHH10]). This paper focuses on *voter control*, a model introduced by Bartholdi et al. [BTT92], where a chair attempts to alter the outcome of an election via changing its structure by deleting, adding, or partition of voters.

To the best of our knowledge, all previous work on control (see, e.g., [BTT92, HHR07, FHHR09, HHR09, ENR09, EPR11]) takes for granted that the chair has full knowledge of all the voters’ preferences and that all votes are cast simultaneously. However, in many settings voters vote sequentially and the chair’s task in such a setting may often be quite different: Knowing only the already cast votes but not the future ones, the chair must decide *online* (i.e., in that moment) whether there exists a control action that guarantees success, no matter what votes will be cast later on. We introduce a framework to model *online voter control in sequential elections*. Our approach is inspired by the area of “online algorithms” [BE98], algorithms running and performing computational actions based only on the input data seen thus far.

In our framework of online voter control, the chair’s task stated above is based on a “maxi-min” idea (although here, due to the time effects, that can involve more than two quantifiers), a typical online-algorithmic theme; in that framing of the chair’s task we are following the approach used for online manipulation of [HHR12a]. Note that another central online-algorithmic theme, a strictly numerical ratio approach to so-called “competitive analysis,” does not apply here, because voting is commonly based on an ordinal notion of preferences, which makes competitive ratios inappropriate in our setting. Sequential (or otherwise “dynamic”) voting has been studied in other contexts as well, e.g., from a game-theoretic perspective as “Stackelberg voting games” [XC10] (see also [DE10]), or using an axiomatic approach [Ten04] or Markov decision processes [PP11]. None of this work has considered the issue of voter control or has applied methods of online algorithms.

2 Preliminaries

We assume familiarity with complexity-theoretic notions such as the complexity classes P, NP, coNP, and PSPACE, the polynomial-time many-one reducibility (\leq_m^P), and with \leq_m^P -hardness and \leq_m^P -completeness for a complexity class. A standard NP-complete problem is the satisfiability problem (SAT) from propositional logic, a standard coNP-complete problem is the tautology problem, and the quantified boolean formula problem (QBF) is a standard PSPACE-complete problem.

Voter Control Types in Simultaneous Elections

A pair (C, V) is called a (*standard or simultaneous*) *election* if C is a set of candidates and V a list of voters that all have cast their votes simultaneously. We assume that each vote in V has the form (v, p) , where v is the name of this voter and p is v ’s (total) preference order over C . For example, if $C = \{c, d, e\}$ then $(\text{Bob}, d > e > c) \in V$ indicates that Bob (strictly) prefers d to e and e to c .

Bartholdi et al. [BTT92] introduced the *standard types of (constructive) voter control in simultaneous elections* as follows. Let \mathcal{E} be a given election system. In *control by deleting voters* (\mathcal{E} -CCDV), given an election (C, V) , a distinguished candidate $c \in C$, and a nonnegative integer $k \leq \|V\|$, we ask whether we can delete at most k voters from V such that c is an \mathcal{E} winner of the resulting election. In *control by adding voters* (\mathcal{E} -CCAV), we are given a candidate set C , a list V of registered voters with

preferences over C , a list V' of as yet unregistered voters with preferences over C , a distinguished candidate $c \in C$, and a nonnegative integer $k \leq \|V'\|$, and the question is whether we can add to V at most k voters from V' such that c is an \mathcal{E} winner of the resulting election. Finally, in *control by partition of voters*, we are given an election (C, V) and a distinguished candidate $c \in C$, and we ask whether V can be partitioned into two sublists, V_1 and V_2 , such that c is an \mathcal{E} winner of the election $(C, W_1 \cup W_2)$, where W_i for $i \in \{1, 2\}$ is the (possibly empty) set of winners of subelection (C, V_i) that have survived the tie-handling rule used. Of the two tie-handling models introduced by Hemaspaandra et al. [HHR07] we focus on the *ties-promote (TP)* model only, where all winners of a subelection proceed to the runoff, as that model fits more naturally the nonunique-winner model in which we will define our online control problems. The resulting problem is denoted by \mathcal{E} -CCPV.

The destructive variants of these three problems, denoted by \mathcal{E} -DCDV, \mathcal{E} -DCAV, and \mathcal{E} -DCPV, are obtained by requiring that the distinguished candidate c is *not* a winner of the election resulting from the control action at hand [HHR07].

Online Voter Control in Sequential Elections

We study *online voter control in sequential elections*, where we assume that the voters vote in order, one after the other, over all candidates. If u is the current voter and C the given candidate set, an *election snapshot for C and u* is specified by a triple $V = (V_{<u}, u, V_{u<})$, where the voters in $V_{<u}$ have already cast their votes, each a preference order over C , now it is u 's turn to cast a vote, and the future voters in $V_{u<}$ will cast their votes in the order listed.

We now define our notions of online voter control for the standard voter control types stated above, and the related problems. They all will start from a basic *online voter control setting* (an *OVCS*, for short), augmented by appropriate additional information according to the control type at hand. A basic OVCS (C, u, V, σ, d) consists of a set C of candidates, the current voter u , an election snapshot for C and u , the chair's preference order σ on C , and a distinguished candidate $d \in C$. Let \mathcal{E} be a given election system and let $W_{\mathcal{E}}(C, V)$ denote the \mathcal{E} winner set of (standard) election (C, V) . For each online voter control type we will define, the question the chair faces is: Is it possible to decide whether or not to exert the action of this control type to the current voter u (e.g., whether or not to delete u) such that, no matter what votes the remaining voters after u wish to cast, the chair's goal can be reached by the current decision regarding u and by using the chair's future decisions of this type (if any), each being made using the chair's then-in-hand knowledge about what votes will have been cast by then? By *the chair's goal* we mean either to ensure $W_{\mathcal{E}}(C, V') \cap \{c \mid c \geq_{\sigma} d\} \neq \emptyset$ for each possible ultimate election (C, V') (i.e., V' is one possible vote list resulting from the control type at hand after all decisions have been made by the chair and all voters have cast their votes) in the constructive case, or to ensure that $W_{\mathcal{E}}(C, V') \cap \{c \mid d \geq_{\sigma} c\} = \emptyset$ in the destructive case. Note, in particular, that due to the conditions $W_{\mathcal{E}}(C, V') \cap \{c \mid c \geq_{\sigma} d\} \neq \emptyset$ and $W_{\mathcal{E}}(C, V') \cap \{c \mid d \geq_{\sigma} c\} = \emptyset$ that define the chair's goal, we adopt the *nonunique-winner* model in defining our problems. So, to formally define our problems, it remains to specify for each control type the information by which the basic OVCS is augmented. What kind of decisions the chair is to make in the course of a sequential election will always be clear from the control type at hand (e.g., whether or not to delete a voter in "online control by deleting voters").

Let $B = (C, u, V, \sigma, d)$ be a given basic OVCS. For *online control by deleting voters*, B is augmented by the following additional information: A nonnegative integer k (the deletion upper bound); for each voter v before u , there is a flag saying if v was deleted and the vote cast by v (if not deleted)—at most

k voters can be marked as deleted for the input to be syntactically legal; a vote to cast for the current voter u (if u is not to be deleted). We denote these problems by online- \mathcal{E} -CCDV (constructive) and online- \mathcal{E} -DCDV (destructive).

For *online control by adding voters*, B is augmented by the following additional information: A nonnegative integer k (the addition upper bound); each voter v in V has a flag saying if v is unregistered (i.e., can be added) or registered— u must be unregistered; each voter v before u has another flag saying if v was added—at most k voters have that flag set in any syntactically legal input; the vote cast is given for each registered or added unregistered voter before u , and also u 's potential vote (if u is to be added). We denote these problems by online- \mathcal{E} -CCAV (constructive) and online- \mathcal{E} -DCAV (destructive).

For *online control by partition of voters*, B is augmented by the following additional information: Each voter v before u has a flag saying which part of the partition v was assigned to (“left” or “right”), and the vote cast by v and also u 's vote is given. We denote these problems by online- \mathcal{E} -CCPV (constructive) and online- \mathcal{E} -DCPV (destructive), as we focus on the ties-promote (TP) rule. This is the right choice for the nonunique-winner model, which itself is here more natural than the unique-winner model, since our online control problems are defined via *upper-cuts* (“make d or a better candidate win,” in the constructive case) or *lower-cuts* (“make sure that neither d nor a worse candidate wins,” in the destructive case).

A natural worry about our maxi-min approach to online voter control is that it is always possible that all the future voters are hostile to one's goals. And in that case, one may be, depending on the election system, powerless to reach one's goal in the worst case, and so the maxi-min outcome is easily seen to be failure to reach one's goal. Although this worry exists in a weaker form for online manipulation and online bribery, since for those if one is allowed almost no vote-changing one is in many cases obviously in trouble, at least in those settings one can do whatever one wants to those votes one does manipulate or bribe. In control, one doesn't get to set the value of a single vote, and that is pretty extreme.

This is a valid worry, but there are some things that keep it in perspective. Primarily, our paper is trying to find out the very greatest complexity that control can ever have (when restricted to election systems having polynomial-time winner problems). And so we can look at election systems that sidestep the above worry, due to their properties simply not matching the intuition above, which is that we are using an election system in which a lot of bad-for-us votes result in a bad-for-us output. In effect, we are seeking to understand the limits of behavior, in order to set a bounding box on the behaviors that can be realized. Of course, for many natural election systems, the effect mentioned in the previous paragraph will hold, and for many inputs that fact can be exploited to help achieve polynomial-time algorithms for the control problem; indeed, in this paper itself, we give examples of achieving polynomial-time algorithms for a natural system: plurality. Of course, problems may start with some votes already cast, and this may itself potentially make for interesting “endgame” decision issues, as may issues involving weighted votes. Also, we very much hope further studies will be conducted (by us or by others) employing a range of models, including ones beyond maxi-min.

3 General Upper and Lower Bounds

Theorem 1 *For each election system \mathcal{E} with a polynomial-time winner problem,¹ online- \mathcal{E} -CCDV, online- \mathcal{E} -DCDV, online- \mathcal{E} -CCAV, online- \mathcal{E} -DCAV, online- \mathcal{E} -CCPV, and online- \mathcal{E} -DCPV are each in PSPACE.*

PROOF. The upper bounds follow from the observation that each of these problems can be solved by an alternating Turing machine in polynomial time, and thus by a deterministic polynomial-space Turing machine, by the characterization due to Chandra et al. [CKS81]. \square

Theorem 2 settles all general (i.e., regarding any voting system for which winner determination is easy) upper bounds for our online voter control problems. We now turn to their lower bounds.

3.1 Control by Deleting and by Adding Voters

Theorem 2 *There exist election systems \mathcal{E} and \mathcal{E}' with polynomial-time winner problems such that online- \mathcal{E} -CCDV, online- \mathcal{E} -CCAV, online- \mathcal{E}' -DCDV, and online- \mathcal{E}' -DCAV are PSPACE-complete, even if limited to only two candidates.*

PROOF. Let (C, V) be an election. We define election system \mathcal{E} as follows. \mathcal{E} interprets—in some fixed, natural encoding—the lexicographically least candidate name in C as a boolean formula, Φ , whose variable names must be the strings $x_1, x_2, \dots, x_{2\ell}$ for some ℓ , where $x_{2\ell}$ actually appears in Φ (the other variables don’t have to; no variables other than $x_1, x_2, \dots, x_{2\ell}$ are allowed). If these syntactic requirements fail to hold, everyone loses in \mathcal{E} . Otherwise, if any two voters in V have the same name, everyone loses in \mathcal{E} . Otherwise, order the voters in V lexicographically by name of the voter, and let v_1, v_2, \dots, v_z be the voter names in this order. If $z < 2\ell$ or if there are less than two candidates, everyone loses in \mathcal{E} . Otherwise, if for some odd i , $1 \leq i \leq 2\ell - 1$, the two lowest order bits of v_i are not 00 or 01, or if for some even i , $2 \leq i \leq 2\ell$, the two lowest order bits of v_i are not 10 or 11, everyone loses in \mathcal{E} . Otherwise, assign the variables of $\Phi(x_1, x_2, \dots, x_{2\ell})$ as follows. For each odd i , $1 \leq i \leq 2\ell - 1$, set x_i to *true* if the two lowest order bits of v_i are 01, and set x_i to *false* otherwise (i.e., the two lowest order bits of v_i are 00). For each even i , $2 \leq i \leq 2\ell$, set x_i to *true* if the name of the least preferred candidate in the vote of v_i is lexicographically less than the name of the next to least preferred candidate in the vote of v_i , and set x_i to *false* otherwise. If this assignment satisfies Φ , everyone wins in \mathcal{E} , and otherwise everyone loses. This ends the specification of \mathcal{E} . Since a boolean formula whose variables have all been assigned can be evaluated in polynomial time, \mathcal{E} has a polynomial-time winner problem.

By Theorem 1, online- \mathcal{E} -CCDV is in PSPACE. To show PSPACE-hardness of online- \mathcal{E} -CCDV, we \leq_m^P -reduce the PSPACE-complete problem QBF', a variant of QBF, to it. QBF' is the set of boolean formulas of the form $F(x_1, x_2, \dots, x_{2\ell})$, for some ℓ , such that the variable $x_{2\ell}$ appears in F , all variables appearing in F are from the variable name collection “ x_1 ”, “ x_2 ”, \dots , “ $x_{2\ell}$ ”, and

$$(\exists b_1)(\forall b_2) \cdots (\exists b_{2\ell-1})(\forall b_{2\ell}) [F(x_1 := b_1, x_2 := b_2, \dots, x_{2\ell} := b_{2\ell}) \text{ evaluates to true}],$$

where $b_i \in \{0, 1\}$ and $x_i := b_i$ means that variable x_i is set to *true* if $b_i = 1$, and is set to *false* if $b_i = 0$, for $1 \leq i \leq 2\ell$.

¹The statement of Theorem 1 holds even for election systems whose winner problems are in PSPACE.

Let $F(x_1, x_2, \dots, x_{2\ell})$ be a given instance of QBF' , where $x_{2\ell}$ explicitly appears in F . (If our input is syntactically incorrect, we map it to a fixed no-instance of $\text{online-}\mathcal{E}\text{-CCDV}$.) We construct from F an instance of $\text{online-}\mathcal{E}\text{-CCDV}$, consisting of a basic OVCS (C, u, V, σ, d) , augmented by the additional information of online control by deleting voters, as follows. Define $C = \{a, b\}$, where a encodes F (in our fixed, natural encoding of boolean formulas) and b is the string lexicographically immediately following a ; the current voter is $u = v_1$; V will be specified below; the chair's preference order is $a >_\sigma b$; for specificity, we let $d = a$ be the distinguished candidate (though that does not matter, as all candidates win or all lose in \mathcal{E}); the deletion limit is $k = \ell$; and a vote $a > b$ to cast for u if not deleted (again, the vote doesn't matter, as $u = v_1$ will specify an assignment to x_1 by her name, not by her vote). There are $(3/2) \cdot 2\ell = 3\ell$ voters in V such that the name of the i th voter, v_i , is the binary string $u_i w_i$, where u_i is the binary representation of i and $w_i = 00$ if $i \equiv 1 \pmod 3$, $w_i = 01$ if $i \equiv 2 \pmod 3$, and $w_i = 10$ if $i \equiv 0 \pmod 3$, $1 \leq i \leq 3\ell$. This completes the description of our \leq_m^P -reduction from QBF' to $\text{online-}\mathcal{E}\text{-CCDV}$, which clearly can be computed in polynomial time.

We claim that $F \in \text{QBF}'$ if and only if the chair's goal can be reached by at most k deletions of voters. Why? By the definition of \mathcal{E} , everyone loses unless our $k = \ell$ deletions are used on *exactly one* of v_{3i-2} and v_{3i-1} , for each i , $1 \leq i \leq \ell$. No v_{3i} , $1 \leq i \leq \ell$, can be deleted if there is to be a winner. And the “exactly one of v_{3i-2} and v_{3i-1} ” choices, $1 \leq i \leq \ell$, specify an assignment of truth values to the odd-numbered variables: For each i , $1 \leq i \leq \ell$, x_{2i-1} is set to *true* if v_{3i-2} is deleted and v_{3i-1} is not, and is set to *false* if v_{3i-1} is deleted and v_{3i-2} is not. On the other hand, for each i , $1 \leq i \leq \ell$, the truth value of x_{2i} is specified by the *vote* of voter v_{3i} , since after these ℓ deletions, v_{3i} will be the $2i$ th voter name in the lexicographic order. It follows that the chair's goal can be reached by at most k deletions of voters if and only if $(\exists b_1)(\forall b_2) \dots (\exists b_{2\ell-1})(\forall b_{2\ell}) [F(x_1 := b_1, x_2 := b_2, \dots, x_{2\ell} := b_{2\ell})]$ evaluates to *true*, which is true if and only if $F \in \text{QBF}'$.

PSPACE-hardness of $\text{online-}\mathcal{E}\text{-CCAV}$ for the election system \mathcal{E} defined above can be shown via essentially the same \leq_m^P -reduction from QBF' . The only difference is that we now map the given QBF' instance F to an instance of $\text{online-}\mathcal{E}\text{-CCAV}$, which is defined exactly as the $\text{online-}\mathcal{E}\text{-CCDV}$ instance constructed above, except that all voters v_i with $i \equiv 0 \pmod 3$ are specified as registered voters, and all other voters are unregistered. The correctness argument is analogous.

The destructive cases can be shown analogously, by modifying the election system \mathcal{E} defined above as follows, yielding our modified system \mathcal{E}' : Whenever everyone loses (wins) in \mathcal{E} , everyone wins (loses) in \mathcal{E}' . It follows from Theorem 1 and the above \leq_m^P -reduction from QBF' that $\text{online-}\mathcal{E}'\text{-DCDV}$ and $\text{online-}\mathcal{E}'\text{-DCAV}$ are both PSPACE-complete. \square

For control by deleting or adding voters, the deletion or addition limit k is part of the problem instance. Now, we consider restrictions of these problems in which this limit is bounded by a constant. For a given election system \mathcal{E} and a fixed k , let $\text{online-}\mathcal{E}\text{-CCDV}[k]$ be the restriction of $\text{online-}\mathcal{E}\text{-CCDV}$ to those inputs whose deletion limit is at most k , and define the problem variant $\text{online-}\mathcal{E}\text{-CCAV}[k]$ analogously. We now show that this change in the definition brings the complexity of these problems from PSPACE down to coNP.

Theorem 3 *For each $k \geq 0$, the following hold:*

1. *For each election system \mathcal{E} with a polynomial-time winner problem, both $\text{online-}\mathcal{E}\text{-CCDV}[k]$ and $\text{online-}\mathcal{E}\text{-CCAV}[k]$ are in coNP.*

2. *There exists an election system \mathcal{E} with a polynomial-time winner problem such that $\text{online-}\mathcal{E}\text{-CCDV}[k]$ and $\text{online-}\mathcal{E}\text{-CCAV}[k]$ are coNP-complete, even if limited to only two candidates.*

PROOF SKETCH. For the first part, for k additions or deletions there are only at most $\|V\|^k$ or $\binom{\|V\|}{k}$ ways to do those, so this can easily be done by a polynomial-time disjunctive truth-table reduction (see [LLS75] for the definition) to a coNP problem, and so $\text{online-}\mathcal{E}\text{-CCDV}[k]$ and $\text{online-}\mathcal{E}\text{-CCAV}[k]$ are easily seen to be in coNP.

For the second part, even for $k = 0$ (and in effect so for all k , as those have within them $k = 0$ as subcases we can map to) $\text{online-}\mathcal{E}\text{-CCDV}[k]$ is easily shown to be coNP-hard, by a \leq_m^P -reduction from the coNP-complete tautology problem. Similarly to the proof of Theorem 2 we use the lexicographically least candidate name to be a proposed tautology and we use the voters as tests of various assignments to it (if the assignment satisfies, everyone wins). So the problem can force the chair's top choice (candidate a , see the proof of Theorem 2) to win exactly if the formula is a tautology. Recall from this proof that this reduction has only two candidates.

The proof sketch for $\text{online-}\mathcal{E}\text{-CCAV}[k]$ is similar. The first (and current) voter in our reduction is unregistered (but with $k = 0$ she obviously cannot be added), and the remaining voters are testing assignments to a proposed tautology and we have only two candidates, just as in the above proof sketch for $\text{online-}\mathcal{E}\text{-CCDV}[k]$. \square

3.2 Control by Partition of Voters

Theorem 4 *There exist election systems \mathcal{E} and \mathcal{E}' , whose winner problems can be solved in polynomial time, such that $\text{online-}\mathcal{E}\text{-CCPV}$ and $\text{online-}\mathcal{E}'\text{-DCPV}$ are PSPACE-complete, even if limited to only two candidates.*

PROOF. This proof is similar in flavor to the proof of Theorem 2, but since we now handle control by partition of voters, there are some decisive differences.

The election system \mathcal{E} as now defined as follows.

Case 1: There is a candidate named RoundOne, and no voter is named Marker. In this case, everyone loses.

Case 2: There is a candidate named RoundOne and a vote named Marker. In this case, interpret—in our fixed, natural encoding—the lexicographically least candidate not named RoundOne as a boolean formula, Φ , whose variable names must be the strings $x_1, x_2, \dots, x_{2\ell}$ for some ℓ , and $x_{2\ell}$ must actually appear in Φ (the others do not have to, but no variable other than $x_1, x_2, \dots, x_{2\ell}$ can appear in Φ). If this candidate is not of the required syntactic form, exactly RoundOne wins. If the candidate set does not consist of exactly RoundOne and the above candidate, then exactly RoundOne wins. If the voter list consists of exactly $2\ell + 1$ voters such that one voter is named Marker, one voter is named v_1^{yes} or v_1^{no} , one voter is named v_2 , one voter is named v_3^{yes} or v_3^{no} , \dots , one voter is named $v_{2\ell-1}^{\text{yes}}$ or $v_{2\ell-1}^{\text{no}}$, and one voter is named $v_{2\ell}$, where all subscripts are given in binary, then assign the 2ℓ variables of Φ as follows. (If the voter list is not exactly that then exactly RoundOne wins.) For each odd i , $1 \leq i \leq 2\ell - 1$, set x_i to *true* if there is a voter named

v_i^{yes} and to *false* if there is a voter named v_i^{no} . For each even i , $2 \leq i \leq 2\ell$, set x_i to *true* if the voter named v_i has the property that in her preference order RoundOne is the top choice, and otherwise set x_i to *false*. If this assignment makes Φ *true*, then the candidate not named RoundOne is the only winner, otherwise (exactly) RoundOne wins.

Case 3: There is no candidate named RoundOne. In this case, everyone wins.

This ends the specification of \mathcal{E} . Clearly, \mathcal{E} has a polynomial-time winner problem, since it is just evaluating a fully specified and assigned boolean formula, and doing various syntactic checks.

Our online control by partition of voters problems are all in PSPACE by Theorem 1. To prove PSPACE-hardness, we again \leq_m^p -reduce from the PSPACE-complete problem QBF' defined in the proof of Theorem 2. Let $F(x_1, \dots, x_{2\ell})$ be a given QBF' instance, where $x_{2\ell}$ actually occurs in F . (If our input is syntactically incorrect, then map it to a fixed nonmember of our target problem.) Our candidate set will be $C = \{\text{RoundOne}, a\}$, where a will in her name encode F (without loss of generality, that will not form the string “RoundOne”), a will be our distinguished candidate, our current voter will be $u = \tilde{v}_1$, the chair's preference order will be $a >_\sigma \text{RoundOne}$, and there will be $3\ell + 1$ voters who vote in order $\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_{3\ell+1}$, where \tilde{v}_1 is named Marker, and the remaining voters are named as follows:

voter	\tilde{v}_2	\tilde{v}_3	\tilde{v}_4	\tilde{v}_5	\tilde{v}_6	\tilde{v}_7	\dots	$\tilde{v}_{3\ell-2}$	$\tilde{v}_{3\ell-1}$	$\tilde{v}_{3\ell}$
name	v_1^{yes}	v_1^{no}	v_2	v_3^{yes}	v_3^{no}	v_4	\dots	$v_{2\ell-1}^{\text{yes}}$	$v_{2\ell-1}^{\text{no}}$	$v_{2\ell}$

This ends our statement of the reduction. Why does it work?

If $F \in \text{QBF}'$, then

$$(\exists b_1)(\forall b_2) \dots (\exists b_{2\ell-1})(\forall b_{2\ell}) \quad (1)$$

$$[F(x_1 := b_1, x_2 := b_2, \dots, x_{2\ell} := b_{2\ell}) \text{ evaluates to } \textit{true}],$$

where the $b_i \in \{0, 1\}$ are truth assignments. So the partition that puts Marker and all voters v_i , i even, on one side, say into V_{left} , and for each $v_i^{\text{yes}}/v_i^{\text{no}}$ pair, i odd, follows (1) by putting v_i^{yes} into V_{left} and v_i^{no} into V_{right} if $b_i = 1$, and v_i^{no} into V_{left} and v_i^{yes} into V_{right} if $b_i = 0$ (and crucially note that the preference orders of the v_i , i even, we will have seen in the future can (in the future) effect the future partition choices), will by Case 2 have one first-round election (namely, (C, V_{left})) in which a is the only winner. And in the other first-round election, (C, V_{right}) , by Case 1 everyone, including RoundOne, loses. Thus, only a proceeds to the second-round runoff election, where by Case 3 everyone wins, i.e., our distinguished candidate a wins.

In the other direction, suppose F is syntactically correct, and it is possible by some partition of voters to force “ a or better” (so a) to be a winner. Since RoundOne is in both first-round elections (so Case 3 cannot occur), the only way candidate a can be guaranteed to even survive at least one first-round election is if we can guarantee that Case 2 is satisfied. But that means that $F \in \text{QBF}'$.

Since our reduction can be computed in polynomial time, this shows that online- \mathcal{E} -CCPV is PSPACE-hard.

To show that online- \mathcal{E}' -DCPV is PSPACE-hard, we modify the election system \mathcal{E} defined above as follows, yielding our modified system \mathcal{E}' : Most crucially, Case 2 of the election system description changes to now making everyone lose if Φ evaluates to *true* under the specified assignment, and if Φ

evaluates to *false* (or there is any syntactic problem regarding who is in the voter list) then everyone wins. Case 3 changes to now having everyone lose, and Case 1 stays the same. The \leq_m^P -reduction from QBF' remains the same, except that the chair's preference order will now be reversed to $\text{RoundOne} >_\sigma a$, and with these changes the reduction can be shown to work correctly by arguments analogous to those in the constructive case. \square

The above proof established PSPACE-completeness for constructive and destructive online control by partition of voters in election systems with polynomial-time winner problems, even if there are at most two candidates. Can we make do with one? The following result shows that if we could, then PSPACE would equal $\text{NP} \cap \text{coNP}$.

- Theorem 5** 1. *For each election system \mathcal{E} with a polynomial-time winner problem, the problems online- \mathcal{E} -CCPV and online- \mathcal{E} -DCPV, when inputs are restricted to at most one candidate, are both in NP.*
2. *There exist election systems \mathcal{E} and \mathcal{E}' with polynomial-time winner problems such that the problems online- \mathcal{E} -CCPV and online- \mathcal{E}' -DCPV, even when restricted to one candidate, are both NP-complete.*

PROOF. We give the proof for the destructive case. For the first part, with one candidate, c , every voter has the same preference as her full vote: c . So there is no sequentially revealed information, as in our model we know the voter names (and their order but here that does not matter) as part of our input. So we just in NP can guess every partition of the voters from u , the current voter, onward, and see if one of those meets the chair's destructive goal, " c does not win."

For the second part, membership in NP follows from the first part. As to NP-hardness, let us \leq_m^P -reduce from SAT. The election system, \mathcal{E}' , is defined as follows:

Case 1: If there are two or more candidates, everyone wins.

Case 2: If there is one candidate and that candidate's name gives a syntactically correct boolean formula φ that has, say, k variables, and there are exactly k voters, and if we set the i th variable of φ to *true* exactly if 1 is the lowest order bit of the voter whose name ranks i th in lexicographic order among the voters' names, then φ is satisfied either by that assignment or by the bitwise complemented twin of that assignment, then everyone loses.

Case 3: In all other cases (including syntactical problems), everyone wins.

The reduction $\text{SAT} \leq_m^P \text{online-}\mathcal{E}'\text{-DCPV}$ is defined as follows. Given a boolean formula $F(x_1, \dots, x_k)$, where without loss of generality all variables actually appear in F , we construct an online- \mathcal{E}' -DCPV instance with candidate set $C = \{c\}$, where c encodes F , the voters are named (in binary) $1, 2, \dots, 2k$ and they vote in this order, $u = 1$ is the current voter, the distinguished candidate is c , and the chair's preference order σ is c . Clearly, c can be made not a winner if and only if F is satisfiable. Why?

First, if F is satisfiable then we can determine a satisfying assignment by the partition choices we make among each voter pair $(2i-1, 2i)$, $1 \leq i \leq k$, by choosing exactly one per pair for the right-hand side of the partition, such that the left-hand side of the partition has the bit-wise complement of that

same satisfying assignment. So, by the definition of \mathcal{E}' , c will not be a winner in either first-round subelection, and so will not even be in the final runoff election, which will have zero candidates, and so c will not be a winner.

Second, if c loses, by the election rule that proves that (Case 2 in the definition of \mathcal{E}'), F is satisfiable. \square

Corollary 6 *The following three statements are equivalent:*

1. $\text{PSPACE} = \text{NP} \cap \text{coNP}$.
2. *There exists an election system \mathcal{E} with a polynomial-time winner problem such that online- \mathcal{E} -DCPV is PSPACE-complete even when restricted to one candidate.*
3. *There exists an election system \mathcal{E} with a polynomial-time winner problem such that online- \mathcal{E} -CCPV is PSPACE-complete even when restricted to one candidate.*

PROOF. To show equivalence of the first two statements, suppose $\text{PSPACE} = \text{NP} \cap \text{coNP}$. So $\text{PSPACE} = \text{NP}$. The second statement now follows from the second part of Theorem 5. Conversely, by the hypothesis and the first part of Theorem 5, we have $\text{PSPACE} \subseteq \text{NP}$, so $\text{PSPACE} = \text{NP} \cap \text{coNP}$. Equivalence of the first and the third statement can be proven analogously. \square

The analogues of the destructive cases of both parts of Theorem 5 also hold when “online” is removed, i.e., for the problem \mathcal{E} -DCPV. In contrast, the *constructive non-online* analogue of Theorem 5’s first part can be strengthened to a P upper bound. (Why can we get a P result here but not in Theorem 5? The proof of the following result does not apply if some voters are already committed to sides of the partition—it is assuming (and truly using the fact) that we have full control of where *all* voters go. But in the online setting, the current voter u can be a voter who does *not* come first and so some voters may already be assigned to sides of the partition. And why do we get P for constructive but not destructive? The effect the following proof uses is specific to the constructive case.)

Theorem 7 *For each election system \mathcal{E} with a polynomial-time winner problem, \mathcal{E} -CCPV, when inputs are restricted to at most one candidate, is in P.*

PROOF. For the one candidate to win, she certainly must win the runoff, in which all voters vote. Also, if she does win when all voters vote, then she can easily be made to survive the first round, using the partition structure (V, \emptyset) . It follows from these two observations that constructive (non-online) control by partition of voters is possible if and only if the one candidate wins in the election with voter list V . \square

4 Online Control for Plurality

We have seen in the previous section that online control can be a very hard, namely a PSPACE-complete, problem, even for voting systems whose winners can be determined in polynomial time. In this section, we study online control for plurality voting. In this very simple, yet popular voting

system, every voter gives one point to her most preferred candidate, and all candidates with the most points win. It is known that non-online control by adding and by deleting voters can be done in polynomial time, both in the constructive [BTT92] and in the destructive [HHR07] case. We now show that the corresponding types of online control are also easy.

Theorem 8 *The problems online-plurality-CCDV, online-plurality-CCAV, online-plurality-DCDV, and online-plurality-DCAV are each in P.*

PROOF. For online-plurality-CCDV, let (C, u, V, σ, d) be a given basic OVCS, augmented by the additional information of online control by deleting voters: a deletion upper bound k , for each voter v before u a flag saying if v was deleted and the vote cast by v (if not deleted), where at most k voters can be marked as deleted, and a vote to cast for u (if u is not to be deleted). If d is the chair's bottom choice in σ , we are done, since the input then is trivially in online-plurality-CCDV (unless it is syntactically illegal). If exactly k voters have been marked as already deleted, we can do no more deletions, so u and all later voters go in, and we assume (as this is the most challenging case) that all later voters vote for one particular candidate in $\Lambda_d = \{c \in C \mid c <_\sigma d\}$ that among the candidates in Λ_d has the most first place votes after u is put in, and so we can easily answer the online control question. If less than k voters have been selected already for deletion, then delete u if and only if u 's top choice is a highest scoring (with respect to the voters before u) candidate in $\{c \in C \mid c <_\sigma d\}$. Then assume that all later voters vote for one particular candidate in $\Lambda_d = \{c \in C \mid c <_\sigma d\}$ that among the candidates in Λ_d has the most first place votes after u is put in. And assume we delete as many of those as the deletion amount left (after u) allows. It is easy to see whether this results in “ d or better” being a winner (in which case our algorithm answers “yes”) or not (in which case our algorithm answers “no”).

For online-plurality-CCAV, let (C, u, V, σ, d) be a given basic OVCS, augmented by the additional information of online control by adding voters: an addition upper bound k , for each voter the information of whether she is registered or not, and for each unregistered voter before u the information of whether she has been added or not, the vote of each registered or added voter before u , and u 's potential vote. Again, the question is trivial if d is the chair's bottom choice in σ . Otherwise, we can see what u 's vote is and if k has yet been reached. If k has not been reached yet, we add u if and only if u 's top choice belongs to $\{c \in C \mid c \geq_\sigma d\}$.² And in the worst case all future voters vote for the same member of $\{c \in C \mid c <_\sigma d\}$, which will be one that after u votes has the most first-place votes among those.

The two destructive cases can be handled analogously. The main differences are, in both cases, that the question now is trivial to decide if d is the chair's *top* choice in σ ; in the deleting voters case, that u is to be deleted (provided the deletion limit k has not been reached yet) if and only if u 's top choice is a highest scoring (with respect to the voters before u) candidate in $\{c \in C \mid c \leq_\sigma d\}$; and in the adding voters case, that u is to be added (provided the addition limit k has not been reached yet) if and only if u 's top choice belongs to $\{c \in C \mid c >_\sigma d\}$. And, in both cases, we again assume that all future votes will belong to some particular member of $\{c \in C \mid c \leq_\sigma d\}$ that after u votes has the most first-place votes among those candidates. \square

²Sure enough, u 's top choice could be one of those candidates that end up having only few votes, so adding u could be a wasted addition that will block some future good addition in some vote sequences, but in the worst case all future voters put first a candidate disliked by the chair; so our action is fine within the quantifier structure of the problem.

Non-online control by partition of voters, in the model (ties-promote) we feel is most natural and have adopted in this paper, is known to be NP-complete [HHR07] in both the constructive and destructive cases. In contrast, the corresponding types of online control are both coNP-hard. This implies that these problems cannot be in NP, unless $NP = coNP$, which is considered to be highly unlikely. It remains open whether or not they are in coNP; we conjecture that they are not.

Theorem 9 *online-plurality-CCPV and online-plurality-DCPV are both coNP-hard.*

PROOF. We prove this by a reduction from the complement of the NP-complete problem Hitting Set: Given a set $B = \{b_1, \dots, b_m\}$, a nonempty collection $\mathcal{S} = \{S_1, \dots, S_n\}$ of subsets of B , and a positive integer $k \leq m$, does \mathcal{S} have a hitting set of size at most k , i.e., does there exist a set $B' \subseteq B$ such that $\|B'\| \leq k$ and for all $S_i \in \mathcal{S}$, $S_i \cap B' \neq \emptyset$.

We turn an instance (B, \mathcal{S}, k) of hitting set into the following instance of online partition of voters. The set of candidates is $\{c, w, b_1, \dots, b_m\} \cup A$, where $A = \{a_i \mid 1 \leq i \leq 4mnk + 1\}$. The current voter is u . The votes before u that are on the left side of the partition are exactly the same as the votes before u that are on the right side of the partition. Both sides of the partition consists of the following votes.

- $4nk$ votes $c > w > \dots$, where \dots denotes that the remaining candidates follow in some arbitrary order.
- $4nk$ votes $w > c > \dots$.
- For every i , $1 \leq i \leq n$, $2k$ votes $S_i > c > \dots$, where S_i denotes the candidates in S_i in some arbitrary order.
- For every j , $1 \leq j \leq m$, as many votes $b_j > B - \{b_j\} > c > w > \dots$ as needed to make the score of b_j equal to $4nk - 1$ in this subelection.
- For every i , $1 \leq i \leq 4mnk$, one vote $a_i > c > \dots$ and one vote $a_i > w > \dots$.

Voter u votes $a_{4mnk+1} > w > \dots$. And there are k voters after u . The chair's top choice is c and the chair's bottom choice is w , and the distinguished candidate is c in the constructive case (i.e., for online-plurality-CCPV) and w in the destructive case (i.e., for online-plurality-DCPV). A simple but crucial observation is that no candidate $a \in A$ will ever make it to the final round, since her score in the first round will be at most $2 + k$. If c and w participate in the final round, c gets $8mnk$ points and w gets $8mnk + 1$ points from voters whose top choice was in A . This will ensure that c and w are the only possible winners in the final round.

We will show that \mathcal{S} does not have a hitting set of size at most k if and only if c can always be made a winner in the constructed election, and we will show that \mathcal{S} does not have a hitting set of size k if and only if w can always be made to not be a winner in the constructed election. This will prove the theorem.

First suppose that \mathcal{S} has a hitting set of size at most k . Let B' be a hitting set of size k . B' exists, since $k \leq m$. Let the k voters after u vote such that the top choice of the i th voter is the i th candidate in B' . Then, no matter how we partition the voters, the set of candidates that participate in the final round is $\{c, w\} \cup B'$. The scores in the final round are as follows: (a) $score(c) = 8nk + 8mnk$,

(b) $\text{score}(w) = 8nk + 8mnk + 1$, and (c) for all $b \in B'$, $\text{score}(b) \leq 8mnk - m + k$. It follows that c is not a winner and that w is a winner.

For the converse, note that we can always make sure that the set of candidates in the final round is of the form $\{c, w\} \cup B'$, where $B' \subseteq B$ and $\|B'\| \leq k$, by putting all the voters after u in the same first-round election. If there is no hitting set of size at most k , then B' is not a hitting set. It follows that in the final election the following hold: (a) $\text{score}(c) \geq 8nk + 8mnk + 4k$, (b) $\text{score}(w) \leq 8nk + 8mnk + 1 + k$, and (c) for all $b \in B'$, $\text{score}(b) \leq 8mnk - m + k$. It follows that c is the unique winner of this election. \square

5 Conclusions and Open Questions

Inspired by the maxi-min approach of online algorithms, we studied online voter control in sequential voting. We showed that for suitably constructed election systems with polynomial-time winner problems, the resulting problems can be extremely hard, namely PSPACE-complete, even for only two candidates. For plurality, however, online control by deleting or adding voters is easy, just as in the non-online case. In future work we will study online voter control also for other natural election systems. Another interesting task will be to investigate online control by a typical-case analysis. Although this paper is about online voter control, a sister paper [HHR12b] studies online candidate control, and we also are investigating online bribery.

References

- [BE98] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.
- [BTT92] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Math. and Computer Modeling*, 16(8/9):27–40, 1992.
- [CKS81] A. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *Journal of the ACM*, 26(1), 1981.
- [DE10] Y. Desmedt and E. Elkind. Equilibria of plurality voting with abstentions. In *Proc. ACM-EC'10*, pages 347–356. ACM Press, 2010.
- [ENR09] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Math. Logic Quarterly*, 55(4):425–443, 2009.
- [EPR11] G. Erdélyi, L. Piras, and J. Rothe. The complexity of voter partition in Bucklin and fallback voting: Solving three open problems. In *Proc. AAMAS'11*, pages 837–844. IFAAMAS, 2011.
- [FHH10] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Comm. of the ACM*, 53(11):74–82, 2010.

- [FHHR09] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *JAIR*, 35:275–341, 2009.
- [HHR07] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.
- [HHR09] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Math. Logic Quarterly*, 55(4):397–424, 2009.
- [HHR12a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The complexity of online manipulation of sequential elections. Technical Report arXiv:1202.6655 [cs.GT], Computing Research Repository, arXiv.org/corr/, February 2012.
- [HHR12b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Controlling candidate-sequential elections. Technical Report arXiv:1202.6649 [cs.GT], Computing Research Repository, arXiv.org/corr/, February 2012.
- [LLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–124, 1975.
- [PP11] D. Parkes and A. Procaccia. Dynamic social choice: Foundations and algorithms. Working paper, May 2011.
- [Ten04] M. Tennenholtz. Transitive voting. In *Proc. ACM-EC’04*, pages 230–231. ACM Press, 2004.
- [XC10] L. Xia and V. Conitzer. Stackelberg voting games: Computational aspects and paradoxes. In *Proc. AAAI’10*, pages 697–702. AAAI Press, 2010.